

AI 人工智慧

不同模型預測鳶尾花(Iris)的效果

(with sklearn & KNN & SVM)

組長：07360194 陳維嶸

組員：07360796 沈奕霖

目錄

目的與動機.....	3
實作&比較圖	4 ~ 8
心得.....	9



Iris Versicolor



Iris Setosa



Iris Virginica

目的與動機

- 針對課程上提到的 Keras 深度學習程式庫分類鳶尾花花種，我們決定採用相似內容呈現我們的期末小專案。
- 我們利用 KNN 與 SVM 探討兩個的預判成績哪個比較高。



實作

- 線性 SVM

```
[ ] # 設定 KNN
knn = KNeighborsClassifier()

# 使用 fit 來建置模型, 其參數接收 training data matrix, testing data array 所以進行 y_train.values.ravel() 轉換
knn.fit(X_train, y_train.values.ravel())

KNeighborsClassifier(algorithm='auto', leaf_size=30, metric='minkowski',
                     metric_params=None, n_jobs=None, n_neighbors=5, p=2,
                     weights='uniform')

[ ] # 用 X_test 來預測結果
print(knn.predict(X_test))

[0 0 2 1 0 2 1 2 0 1 2 0 0 1 1 0 2 1 2 0 2 0 1 1 2 0 1 1 2 0 1 2 1 2 1 1
 0 0 0 2 1 2 1 1]

[ ] # 印出LinearSVC預測準確率
print(knn.score(X_test, y_test))

0.9333333333333333
```

```
[ ] # 印出LinearSVC預測準確率
print(knn.score(X_test, y_test))
```

0.9333333333333333

SVM 預測準確率：0.933

- KNN

```
[ ] # 用 X_test 來預測結果
print(knn.predict(X_test))

[0 0 2 1 0 2 1 2 0 1 2 0 0 1 1 0 1 1 2 0 1 0 1 1 2 0 1 1 2 0 1 2 1 2 1 2 1
 0 0 0 2 1 2 2 1]
```

```
▶ # 印出 testing data 預測標籤機率
print(knn.predict_proba(X_test))
```

```
[[1.  0.  0. ]
 [1.  0.  0. ]
 [0.  0.  1. ]
 [0.  0.6 0.4]
 [1.  0.  0. ]
 [0.  0.2 0.8]
 [0.  1.  0. ]
 [0.  0.  1. ]
 [1.  0.  0. ]
 [0.  1.  0. ]
 [0.  0.  1. ]
 [1.  0.  0. ]
 [1.  0.  0. ]
 [0.  1.  0. ]
 [0.  1.  0. ]
 [1.  0.  0. ]
 [0.  0.8 0.2]
 [0.  1.  0. ]
 [0.  0.  1. ]
 [1.  0.  0. ]
 [0.  1.  0. ]
 [1.  0.  0. ]
 [0.  1.  0. ]
 [0.  1.  0. ]
 [0.  0.2 0.8]
 ...
 [0.  0.2 0.8]
 [1.  0.  0. ]
 [0.  1.  0. ]
 [0.  0.8 0.2]
 [0.  0.  1. ]
 [1.  0.  0. ]
 [0.  1.  0. ]
 [0.  0.  1. ]
 [0.  1.  0. ]
 [0.  0.  1. ]
 [0.  0.6 0.4]
 [0.  0.  1. ]
 [0.  0.6 0.4]
 [1.  0.  0. ]
 [1.  0.  0. ]
 [1.  0.  0. ]
 [0.  0.  1. ]
 [0.  1.  0. ]
 [0.  0.  1. ]
 [0.  0.  1. ]
 [0.  1.  0. ]]
```

KNN 預測準確率：0.955

(優於 SVM 的 0.933)

```
[ ] # 印出KNN預測準確率
print(knn.score(X_test, y_test))
```

```
0.9555555555555556
```

比較圖

- KNN1：鄰近值個數為 3、SVM1（線性）
 - KNN2：鄰近值個數為 5、SVM2（gamma=10）
 - KNN3：鄰近值個數為 7、SVM3（ gamma=100）
- 跑 10 次的準確度

```
[66] #隱藏warning
      from warnings import filterwarnings
      filterwarnings('ignore')

      #執行且繪圖
      loop_test()
```

Test time:10
SVM1平均準確度: 0.9577777777777777
SVM1最大準確度: 1.0
SVM1最小準確度: 0.9111111111111111

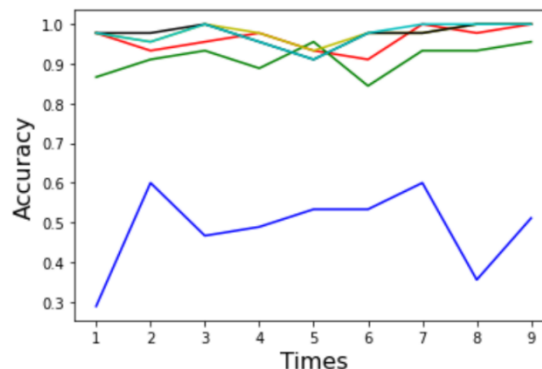
SVM2平均準確度: 0.9155555555555555
SVM2最大準確度: 0.9555555555555556
SVM2最小準確度: 0.8444444444444444

SVM3平均準確度: 0.49555555555555547
SVM3最大準確度: 0.6
SVM3最小準確度: 0.28888888888888886

KNN1平均準確度: 0.9733333333333334
KNN1最大準確度: 1.0
KNN1最小準確度: 0.9333333333333333

KNN2平均準確度: 0.9711111111111113
KNN2最大準確度: 1.0
KNN2最小準確度: 0.9111111111111111

KNN3平均準確度: 0.9733333333333334
KNN3最大準確度: 1.0
KNN3最小準確度: 0.9111111111111111



- 跑 100 次的準確度

```
#隱藏warning
from warnings import filterwarnings
filterwarnings('ignore')

#執行且繪圖
loop_test()
```

```
Test time:100
SVM1平均準確度: 0.95399999999999992
SVM1最大準確度: 1.0
SVM1最小準確度: 0.8666666666666667

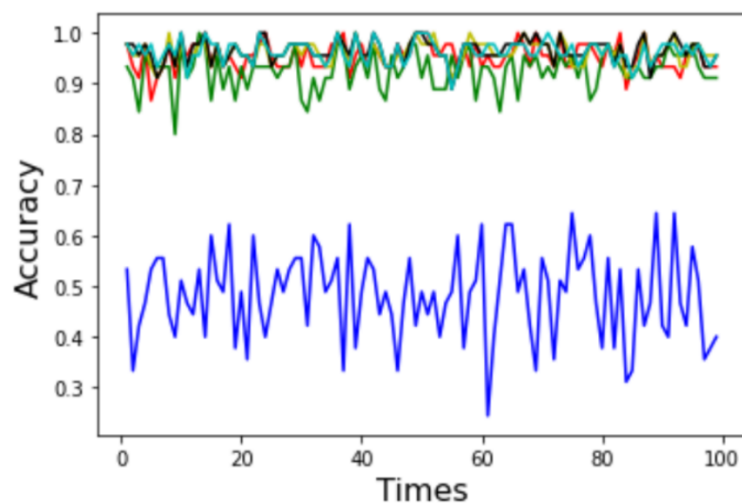
SVM2平均準確度: 0.92511111111111101
SVM2最大準確度: 1.0
SVM2最小準確度: 0.8

SVM3平均準確度: 0.48333333333333345
SVM3最大準確度: 0.64444444444444445
SVM3最小準確度: 0.24444444444444444

KNN1平均準確度: 0.96088888888888879
KNN1最大準確度: 1.0
KNN1最小準確度: 0.91111111111111111

KNN2平均準確度: 0.96466666666666655
KNN2最大準確度: 1.0
KNN2最小準確度: 0.88888888888888888

KNN3平均準確度: 0.96288888888888872
KNN3最大準確度: 1.0
KNN3最小準確度: 0.88888888888888888
```



- 跑 1000 次的準確度

```
#隱藏warning
from warnings import filterwarnings
filterwarnings('ignore')

#執行且繪圖
loop_test()
```

```
Test time:1000
SVM1平均準確度: 0.9535333333333252
SVM1最大準確度: 1.0
SVM1最小準確度: 0.8

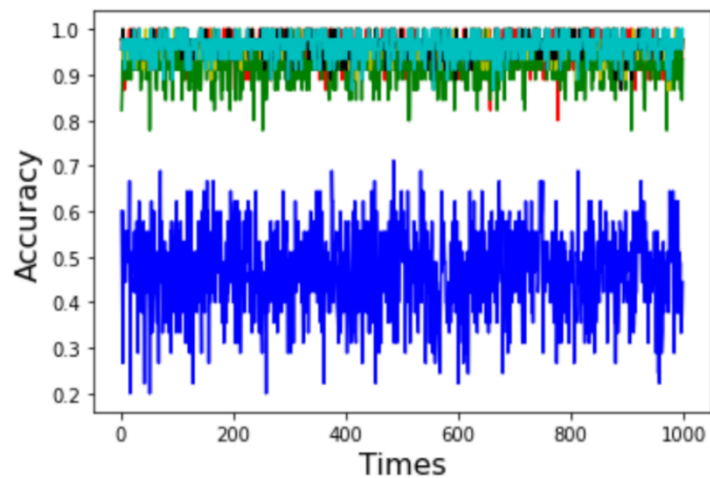
SVM2平均準確度: 0.9265999999999955
SVM2最大準確度: 1.0
SVM2最小準確度: 0.7777777777777778

SVM3平均準確度: 0.466777777777778
SVM3最大準確度: 0.7111111111111111
SVM3最小準確度: 0.2

KNN1平均準確度: 0.9604222222222102
KNN1最大準確度: 1.0
KNN1最小準確度: 0.8666666666666667

KNN2平均準確度: 0.9632444444444334
KNN2最大準確度: 1.0
KNN2最小準確度: 0.8666666666666667

KNN3平均準確度: 0.9642666666666564
KNN3最大準確度: 1.0
KNN3最小準確度: 0.8666666666666667
```



心得

➤ 陳維嶸

在這堂人工智慧中，老師提供了很多線上的資源，這些資料讓我們了解 KNN 和 SVM 等的計算與演算法。

在課堂中我們也學習到很多 lisp 的運算方法，現在如果去書店也可以找到 lisp 相關的資料。我希望老師在學期結束之後不要把資料都關掉，因為裡面的資料非常的具有參考價值。有一個關於 keras 的資料，我覺得這個資料對於我目前的專題研究有非常高的研讀價值。

雖然還有很多套件老師沒有讓我們實作，我想這也是因為學期太短了，沒辦法涉略那麼多套件，不過老師提到的東西，也在未來幫我們鋪了不少的基礎。希望老師可以在未來跟同學介紹目前火紅的 Tensorflow 和 yolo 套件，我想這些東西對於同學的未來會有非常大的幫助，我本身也蠻想學習 Tensorflow 的，希望未來有相關課程可以修。

➤ 沈奕霖

剛開始在構想時我們就想把上課所學到的東西運用到專案裡面，因此這次的專案我們利用不同的分析方法來預測資料的準確率。透過這次專題，雖然沒辦法將人工智慧和深度學習研究的很透徹，但讓我對這個領域有了更進一步的認識。

僅有短暫的一個學期，能學到的東西有限，而深度學習的威力，不容小覷！希望在未來有更多的機會，可以學到更多深度學習的學問，，同時也能增加自身的競爭力。